

Research Article

A Deep Graph-Embedded LSTM Neural Network Approach for Airport Delay Prediction

Weili Zeng ¹, Juan Li ¹, Zhibin Quan ^{2,3} and Xiaobo Lu³

¹College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

²Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China

³School of Automation, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Weili Zeng; zwlnduaa@nuaa.edu.cn

Received 28 November 2020; Revised 1 March 2021; Accepted 10 March 2021; Published 25 March 2021

Academic Editor: Jinjun Tang

Copyright © 2021 Weili Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the strong propagation causality of delays between airports, this paper proposes a delay prediction model based on a deep graph neural network to study delay prediction from the perspective of an airport network. We regard airports as nodes of a graph network and use a directed graph network to construct airports' relationship. For adjacent airports, weights of edges are measured by the spherical distance between them, while the number of flight pairs between them is utilized for airports connected by flights. On this basis, a diffusion convolution kernel is constructed to capture characteristics of delay propagation between airports, and it is further integrated into the sequence-to-sequence LSTM neural network to establish a deep learning framework for delay prediction. We name this model as deep graph-embedded LSTM (DGLSTM). To verify the model's effectiveness and superiority, we utilize the historical delay data of 325 airports in the United States from 2015 to 2018 as the model training set and test set. The experimental results suggest that the proposed method is superior to the existing mainstream methods in terms of accuracy and robustness.

1. Introduction

With the rapid development of the air transport industry, the contradiction between the rapidly increasing air traffic flow and limited airspace resources has become increasingly prominent, leading to frequent flight delays. According to a statistical report from VariFlight, the actual number of flights departing from airports worldwide in 2019 was about 37.12 million, with an on-time departure rate of 75.58% and an average delay of 26.47 minutes [1]. Flight delays have brought many adverse effects on passengers, airlines, and the civil aviation industry. For passengers, flight delays disrupt the passengers' itinerary and cause great inconvenience. For airlines, flight delays affect the passenger travel experience, so passengers may choose other airlines or transportation methods, resulting in a decline in passenger flow and substantial economic losses to the airline. Besides, since each aircraft has to fly multiple flight legs every day, the delay in

the arrival of the previous flight leg will affect the normality of the subsequent flight leg. In the long run, flight delays will affect the development of the entire civil aviation industry. Therefore, how to reasonably arrange limited resources to deal with delays and reduce the losses caused by delays has become the focus of the academic and civil aviation industries.

In view of the flight delay problem, a large number of researchers have conducted research on it and put forward many effective methods and measures from the strategic and tactical levels. For instance, at the strategic level, optimize the airport surface and airspace structure or adopt advanced collaborative operation concepts at the tactical level to improve the operating efficiency of the entire airport and airspace network and reduce flight delays [2–4]. Besides, after flight delays, effective control measures (such as ground waiting) are taken to reduce the extent and scope of flight delays [5–7]. In this paper, we focus on the problem of

airport delay prediction. Predicting airport delays can not only provide a decision-making basis for airports, airlines, and air traffic control departments to solve delay problems but also issue early warnings to passengers so that passengers can reschedule their itinerary in advance, thereby reducing the loss caused by flight delays.

In the past two decades, researchers have conducted a lot of research on airport delay prediction. Early studies mainly used probability models to estimate the delay distribution [8–11]. Tu et al. [8] developed a probability model based on the expectation-maximization and genetic algorithm to predict the distribution of departure delays at Denver International Airport. Mueller and Chatterji [9] used density functions to simulate departure, route, and arrival delays. The results confirm that the normal distribution is more suitable for characterizing departure delay, while the Poisson distribution can better describe the route and arrival delay. Sridhar and Chen [10] proposed a method to develop a short-term delay prediction model. For the first time, the weather-influenced traffic index (WITI) and the predicted WITI were used for real-time air traffic delay prediction. This research is helpful for traffic flow management to guide flow control decisions and determine strategies to reduce delays, cancellations, and costs during various operations. Similarly, Klein et al. [12] developed a multiple regression model for airport delay prediction using a comprehensive weather-influenced traffic index (WITI) toolset and indicators and used historical airport performance and actual weather/planned traffic data to train the model. Later, it was applied to forecasting and successfully predicted the time and extent of the influence of convective and nonconvective weather throughout the year and the delay caused by it.

With the rapid development of big data mining and machine learning technologies, more and more researchers have turned to research data-driven delay prediction models in recent years [13, 14]. Ding and Li [15] considered the characteristics of airport flight operations and proposed a delay prediction model based on the danger model theory and gray model theory. This method adopted a weighted combination mode based on the estimated occupancy ratio of the mean square error. Rebollo and Balakrishnan [16] proposed to use the random forest algorithm to predict the departure delay of a specific origin-destination pair (or an airport) in the next 2 to 24 hours. They investigated the role of the network delay state in predicting future delays. Zhou et al. [17] constructed a decision tree model for delay prediction from the perspective of airlines. By considering the topological characteristics of the aviation network, the node attributes are combined with the K -means clustering algorithm to classify the busyness of airports and improve the accuracy of delay prediction.

As long as the data are sufficient, neural network models have stronger feature representation capabilities than classic prediction methods such as decision trees and support vector machines [18, 19]. Khanmohammadi et al. [20] introduced a multilevel input layer artificial neural network (ANN) model, which can easily see the relationship between different input variables and output variables. They used this method to predict the delays of arriving flights at JFK

International Airport. The technique is superior to the traditional gradient descent backpropagation ANN model in prediction accuracy and training time. Henriques and Feiteira [21] aimed at the delay prediction problem of Hartsfield–Jackson International Airport, used sampling technology to solve the problem of dataset imbalance, and evaluated and compared the performance of multiple methods such as decision trees, random forests, and multilayer perceptrons. Chandramouleeswaran et al. [22] used public flight data in the United States to propose a method to predict the delay status of airports in the US air transportation network. They designed a network delay metric to reduce the feature space, used the classic neural network model and logistic regression model to build a delay prediction model, and found that the prediction performance is greatly affected by the prediction interval and the delay threshold. Taking Beijing Capital International Airport as a case, Yu et al. [23] proposed a deep belief network method to mine the internal patterns of flight delays and integrated support vector machines with the model to implement supervised fine-tuning within the prediction architecture. In addition to the use of feedforward neural networks (FNNs) to predict delays described above, some researchers have also tried to use recurrent neural networks. Li et al. [24] considered the time-series correlation of flight delay sequences, which is difficult for the FNN to capture this feature, proposed a delay time prediction method based on a long short-term memory model (LSTM), and selected relevant features and divided the delay level. Similarly, Kim et al. [25] investigated the effectiveness of the deep learning models in the air traffic delay prediction tasks. By combining multiple models based on LSTM, an accurate and robust prediction model has been built, enabling an elaborate analysis of the patterns in air traffic delays.

Although some researchers have considered spatial dependence and temporal dependence, no researchers have studied flight prediction from the perspective of graph neural networks. Due to the uneven distribution of airports, each airport has a different number of adjacent airports, and the degree of influence between airports is different. Therefore, from the airport network's perspective to study delay prediction, the data can be regarded as generated from non-Euclidean space. Traditional deep learning methods have achieved great success in representing Euclidean space data, but they do not perform well in non-Euclidean space. From the network's point of view, the graph neural network constructs a graph composed of several nodes and edges. By depicting the relationship between different nodes, it can capture the interdependence between nodes, which can well represent non-Euclidean spatial data.

This article will take airports as nodes and use a directed graph network to construct the relationship between nodes. The weight of an edge represents the degree of influence between two airports. For neighboring airports, the weights between them are measured by the spherical distance, while airports connected by flights are determined according to the number of flight pairs. Furthermore, we regard delay propagation between airports as a diffusion process, use diffusion convolution to capture the delay propagation

characteristics, and integrate it into the deep LSTM neural network model to improve the accuracy of airport delay prediction. The main contributions of this paper are as follows:

- (i) To the authors' knowledge, this paper is the first time to study the problem of airport delay prediction from the perspective of graph neural networks, which can better capture the spatiotemporal dependence of delays between airports.
- (ii) The delay propagation between airports is regarded as the diffusion process of a directed graph. A graph network representation approach based on diffusion convolution is presented to characterize the whole airport network.
- (iii) Our proposed delay prediction model, deep graph-embedded LSTM (DGLSTM), inherits advantages of both graph network and sequence-to-sequence LSTM, thereby improving the accuracy of delay prediction.
- (iv) Taking the US airport network as a case, the effectiveness and superiority of our proposed method are verified through comparative experiments with mainstream delay prediction methods.

The organization structure of this paper is as follows: Section 2 describes the airport delay problem. Section 3 defines the weighted adjacency matrix, including the spatial distance weighted adjacency matrix and the demand weighted matrix. The proposed DGLSTM airport delay prediction model is presented in Section 4. Section 5 takes the United States as a case to verify the method through experiments. Finally, some conclusions and future research directions are summarized in Section 6.

2. Airport Delay Forecasting Problem

This paper studies the problem of airport delay prediction from the perspective of an airport network, using historical delay data of all associated airports to predict the average delay of look-ahead times for all airports. Suppose there are P airports in the entire airport network. Define the airport network as a weighted directed graph $G = (z, V, E, W)$, where z is a global attribute, representing the state of the entire airspace, and can also be regarded as the node of the associated airport; $V = \{v_1, v_2, \dots, v_P\}$ is the set of P airport nodes in the graph; $E = E^d \cup E^s$ is the set of edges, where E^d is the edge set composed of adjacent airports in the space, E^s is the edge set composed of airports connected by the route; and $W \in R^{P \times P}$ represents the weighted adjacency matrix:

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1P} \\ w_{21} & w_{22} & \cdots & w_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ w_{P1} & w_{P2} & \cdots & w_{PP} \end{bmatrix}_{P \times P}, \quad (1)$$

where w_{ij} is the weight between airport v_i and v_j . If there exists an edge between them, then $w_{ij} > 0$; otherwise, $w_{ij} = 0$. Since the airport network is a directed graph, for any node v_i

in the graph, its out-degree d_i^{out} is defined as the sum of out-edge weights, and in-degree d_i^{in} is the sum of in-edge weights. The formulas for calculating the out-degree and in-degree are as follows:

$$\begin{aligned} d_i^{\text{out}} &= \sum_{j=1}^P w_{ij}, \\ d_i^{\text{in}} &= \sum_{j=1}^P w_{ji}. \end{aligned} \quad (2)$$

Using the definition of the out-degree and in-degree of a single node, we can obtain the out-degree matrix $D_{\text{out}} = \text{diag}\{d_1^{\text{out}}, \dots, d_P^{\text{out}}\}$ and the in-degree matrix $D_{\text{in}} = \text{diag}\{d_1^{\text{in}}, \dots, d_P^{\text{in}}\}$ of size $P \times P$. These two matrices are diagonal, and the value of the main diagonal corresponds to each node's degree.

Denote $X = (x_{v_1}, x_{v_2}, \dots, x_{v_P})^T \in R^{P \times M}$ as the input information related to airport delays observed on the graph G and $Y = (y_{v_1}, y_{v_2}, \dots, y_{v_P})^T \in R^{P \times N}$ as the output information, where M represents the number of input signal features (such as departure delay, arrival delay, departure flight volume, arrival flight volume, and airspace status) and N represents the number of output signal characteristics (such as departure delay and arrival delay). Let $X^{(t)}$ and $Y^{(t)}$ represent the input and output graph signals at time t , respectively. Airport delay prediction is to learn a function Ψ on an airport network to map the input graph signal of several historical periods to the output graph signals of several look-ahead times:

$$\Psi: [X^{(t-Q+1)}, \dots, X^{(t)}; G] \longrightarrow [Y^{(t+1)}, \dots, Y^{(t+S)}; G], \quad (3)$$

where Q and S are the number of input and output time steps, respectively.

The difficulty of airport delay prediction lies in the complex spatiotemporal dependence and long-term forecast: (a) the airport's delay has a strong time-domain dependence. For example, the delay of a particular flight at the airport will affect the normal takeoff of subsequent flights and cause the spread of flight delays; (b) the spatial distance between neighboring airports has a robust spatial dependence (Figure 1: airports in the red dotted circle). Usually, adjacent airports share the same airspace resources, such as sectors, routes, or waypoints, and departing flights are easily affected by departing flights from neighboring airports. (c) Airports connected by flights also have a strong spatial correlation (Figure 1: airports connected to CVG by black dotted double arrows). Although the distance between the airports connected by flights is relatively long, the departing flights are easily affected by the destination airport. For example, severe weather at the destination will cause extensive delays in the arrival and departure flights at the destination airport, which will affect the associated airports.

The goal of delay prediction in this paper is to predict the average delay time of multiple periods in the future based on the delay data of multiple periods in the airport network's history. It belongs to a class of problems where both input and

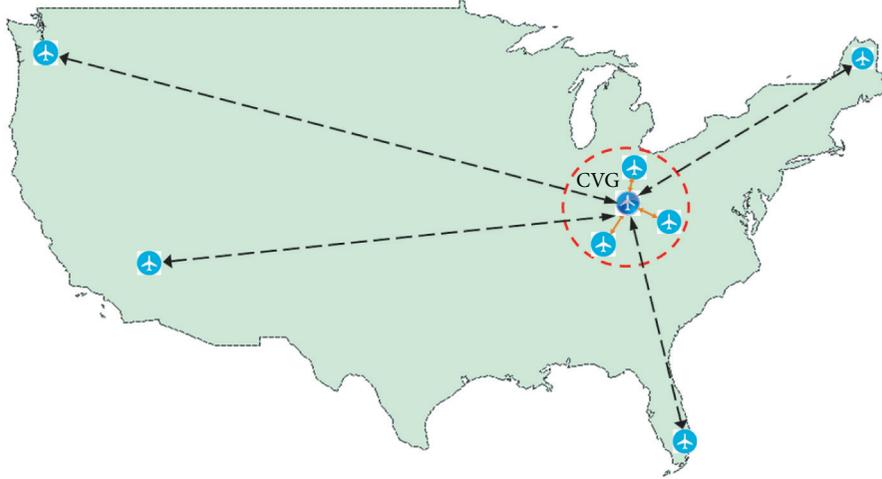


FIGURE 1: Schematic diagram of the airport association relationship, where the red dotted circle represents the local neighborhood centered on Cincinnati International Airport (three-letter code CVG). The black double arrow represents the airport connected to CVG by flights.

output are sequential. Because every airport is affected by other airports and is continuously changing its state, the closer the relationship between airports, the more significant the impact. Therefore, we study the problem of delay prediction from the perspective of the airport network. In other words, in addition to the delay status of the airport itself, the impact of other airports needs to be considered. This paper will integrate the graph convolutional embedding module under the framework of the sequence-to-sequence LSTM neural network [26] and build an airport graph convolutional embedded LSTM prediction model under a unified framework to achieve high-precision prediction of airport flight delays. Figure 2 is a schematic diagram of the DGLSTM neural network delay prediction method, divided into two stages: encoding and decoding. The encoding stage is implemented by the graph representation module and an LSTM module, and another LSTM module implements the decoding stage. The following will introduce how to define the weighted adjacency matrix, the graph network representation module that characterizes spatial dependence, and the sequence-to-sequence LSTM that characterizes temporal relationships.

3. Defining the Weighted Adjacency Matrix

Generally, a graph's network structure varies with the application scenario, and the difference is mainly manifested in the connection mode and weights between nodes. It can be seen from Figure 1 that neighbors of each node in the airport network are divided into two types: airports adjacent in the space and airports connected by air routes. The set of adjacent airport nodes is fixed, and nodes connected by flights are constantly changing over time. Therefore, we first construct the space-weighted adjacency matrix and the demand weighted adjacency matrix separately and then integrate these two adjacency matrices.

3.1. Spatial Distance Weighted Adjacency Matrix. Due to the shared airspace resources (airways, waypoints, sectors, etc.) between adjacent airports in terms of spatial distance, they

will affect each other. Generally speaking, the closer the two airports are, the stronger the correlation between them is, and vice versa. Therefore, for airports with adjacent spatial distances, the two airports' impact is approximately the same; that is, the positive and negative edges have the same weight. This article takes longitude and latitude as the input, calculates the spherical distance between airports, and then uses different kernel functions to define the weights of edges. Commonly used kernel functions include polynomial kernel function, Gaussian kernel function, and sigmoid kernel function. Given the symmetry of the Gaussian kernel function, we use it as the kernel function. For any two airports v_i and v_j , the weight of the airport network edge is calculated according to the following formula:

$$w_{ij} = \begin{cases} \exp\left(-\frac{(\text{dist}(v_i, v_j))^2}{\sigma^2}\right) & \text{if } \text{dist}(v_i, v_j) \leq \varepsilon, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where $\text{dist}(v_i, v_j)$ denotes the spherical distance between v_i and v_j and σ and ε , respectively, represent the standard deviation and the threshold parameter. Obviously, from (3), that is, the spatial distance adjacency matrix is symmetric.

3.2. Demand Weighted Adjacency Matrix. For any two airports connected by flights, the degree of closeness between them can be characterized by incoming and outgoing flights—the more the flights between airports, the stronger the correlation between them, and vice versa. If there are more flights from one airport to another airport, then that airport will have a greater impact on the other airport. For example, from 9 am to 10 am, the number of flights from Nanjing to Beijing is 4, while the number of flights from Beijing to Nanjing is 2. Then, Beijing Airport has a more significant impact on Nanjing Airport than Nanjing Airport has on Beijing Airport. Also, the number of flights at airports connected by flights is continually changing over time.

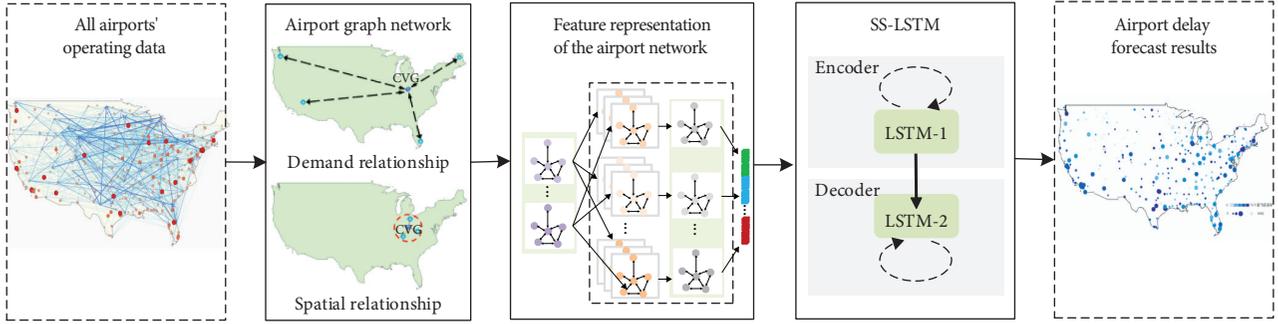


FIGURE 2: Architecture diagram of the DGLSTM airport delay prediction model.

Assume that there are a total of T time intervals (if the hourly average delay is predicted, the whole day can be divided into 24 time intervals). According to the above analysis, for any two airports v_i and v_j , in the t -th time interval, the formula for calculating the weight of airport v_i to airport v_j is

$$w_{ij}^{(t)} = \begin{cases} \frac{O_{v_j \rightarrow v_i}^{(t)}}{\max\{O_{v_j \rightarrow v_i}^{(t)} : t \in \{1, 2, \dots, T\}\}} & \text{if } i \neq j, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where $O_{v_j \rightarrow v_i}^{(t)}$ is the number of flights scheduled to fly from airport v_j to airport v_i . Since the flight plan contains the departure airport, arrival airport, departure time, arrival time, and other flight information, formula (5) can be calculated in advance from the flight plan.

Sections 3.1 and 3.2 discuss how to construct the spatial distance adjacency matrix and the demand adjacency matrix. So, how to integrate these two types of matrices? This article adopts one of the most direct and simple methods, that is, the weighted combination of two types of matrices, as follows:

$$W = \alpha W^{\text{space}} + (1 - \alpha) W^{\text{demand}}, \quad (6)$$

where W^{space} and W^{demand} , respectively, represent the spatial weighting adjacency matrix and the demand weighting adjacency matrix and α is the weighting parameter.

4. DGLSTM Delay Forecasting Model

4.1. Graph Network Representation. The purpose of graph network representation is to extract the feature information of the topological graph. The graph network representation method is divided into two representation methods in the space domain and frequency domain. The difference between them lies in the theoretical basis. The theoretical basis of the spatial representation method is information diffusion theory [27–30], while the theoretical basis of the frequency domain method is convolution theory. Although the theoretical basis is different, some researchers have proved that the two types of techniques are nearly consistent, except that the space they rely on when solving is different. In this

article, we characterize the spatial characteristics of the graph network in the frequency domain.

Graph convolutional network representation uses graph theory to perform convolutional operations on topological graphs to realize the feature representation of graph networks. From the perspective of research ideas, scholars who study graph signal processing first define the Fourier transformation on the graph, then define the convolution operation on the graph, and finally combine with deep learning to propose a graph convolutional neural network. The most critical problem of graph convolutional neural networks is how to construct convolution operators. Since the basis function in the Fourier transform is the characteristic function of Laplacian, the researchers proposed to use the Laplacian matrix (discrete Laplacian) to construct the convolution operator. A large number of studies have shown that the Laplacian operator is a kind of second-derivative operation and a kind of additivity in a large number of information propagation equations.

Researchers have done a lot of research on undirected graph neural networks [31–38], and only a few researchers pay attention to directed graph neural networks [39]. For undirected graph networks, the most representative one was proposed by Bruna and Szlam of New York University [38]. They presented a spectral convolution method to solve the convolution problem of non-Euclidean space. However, this method has poor spatial locality and requires feature decomposition, resulting in high computational complexity. To address the spectral convolution method's deficiencies, Defferrard [31] proposed a Chebyshev spectral convolution method with fewer convolution parameters, and it did not require eigendecomposition and directly uses the Laplacian matrix for transformation, which has better locality. The Laplacian matrix constructed by the spectral convolution method and the Chebyshev spectral convolution method is a positive semidefinite symmetric matrix that can perform eigendecomposition, and the matrix formed by all eigenvectors is orthogonal. The weighted adjacency matrix of the directed graph is usually asymmetric, and the convolution operator constructed based on the Laplacian matrix is no longer applicable in the directed graph. To realize the convolution of the directed graph, Li and Shahabi [39] aimed at the traffic flow prediction problem, modeled the spatial dependence of the traffic flow as a diffusion process in the directed graph, and proposed a diffusion convolution

algorithm that is easy to interpret and efficient to calculate the operator. Given the propagation characteristics of delays in airport networks, this paper generalizes and applies the diffusion convolution operator to the airport network feature representation.

Let $D_O^{-1}W$ and $D_I^{-1}W^T$, respectively, denote the transition matrix of the forward and reverse diffusion process:

$$D_O^{-1}W = \begin{bmatrix} d_1^{\text{out}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_p^{\text{out}} \end{bmatrix}^{-1} \begin{bmatrix} w_{11} & \cdots & w_{1p} \\ \cdots & \ddots & w_{2p} \\ w_{p1} & 0 & w_{pp} \end{bmatrix}, \quad (7)$$

$$D_I^{-1}W^T = \begin{bmatrix} d_1^{\text{in}} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & d_p^{\text{in}} \end{bmatrix}^{-1} \begin{bmatrix} w_{11} & \cdots & w_{1p} \\ \vdots & \ddots & \vdots \\ w_{p1} & \cdots & w_{pp} \end{bmatrix}^T.$$

For the directed graph signal $X = (X_1, \dots, X_M) \in R^{P \times M}$, k -step diffusion convolution of the m -th feature X_m is defined as

$$\begin{aligned} f_\theta * X_m &= \left(\theta_{0,1} (D_O^{-1}W)^0 + \theta_{0,2} (D_I^{-1}W^T)^0 \right) X_m + \cdots \\ &\quad + \left(\theta_{K-1,1} (D_O^{-1}W)^{K-1} + \theta_{K-1,2} (D_I^{-1}W^T)^{K-1} \right) X_m \\ &= \sum_{k=0}^{K-1} \left(\theta_{k,1} (D_O^{-1}W)^k + \theta_{k,2} (D_I^{-1}W^T)^k \right) X_m, \end{aligned} \quad (8)$$

where f_θ is a filter with parameter $\theta \in R^{K \times 2}$.

Assume that the diffusion convolution has $L + 1$ layers, where the 0th layer is the input layer and the l -th ($1 \leq l \leq L$) layers are convolutional layers. Figure 3 shows a schematic representation of the graph feature with two convolutional layers. For any convolutional layer $l \in [1, \dots, L]$, the output of the l -th ($1 \leq l \leq L$) convolution layer is calculated using the following equation:

$$H_j^{(l)} = \sigma \left(\sum_{m=1}^{M_{l-1}} f_\theta^l * H_m^{(l-1)} \right), \quad (9)$$

where M_{l-1} denotes the number of features in the $(l-1)$ -th layer; $H_m^{(l)} \in R^{P \times M_{l-1}}$ is the m -th graph signal in the l -th layer, where $H_m^{(0)} = X_m$; f_θ^l is the convolution filter in the l -th layer; and $\sigma(\cdot)$ is the activation function, such as ReLU function and sigmoid function.

4.2. Modeling Temporal Dependency. In addition to being easily affected by the associated airport, flight delays are also easily affected by the airport's previous flights. If the current flight is delayed significantly, it will affect the departure of subsequent flights and cause the delay of subsequent flights. In other words, flight delays have a specific time-domain dependence. Therefore, when constructing a flight delay prediction model, it is also necessary to consider the time-dependent relationship to realize the feature representation in the time domain. LSTM recurrent neural network has a strong ability to represent long-term and short-term

information and has been widely used in various fields such as energy, finance, and transportation [40–44]. To achieve multistep prediction, we use a sequence-to-sequence LSTM model to capture time-domain features. The model consists of two processes: encoding and decoding (Figure 4). The encoding stage takes the results of the graph network representation module for each time interval as the input, and the decoding stage takes the final output of the encoding stage as the initial input. Encoding and decoding adopt different LSTM processing units and deep network structures.

For convenience, whether it is the encoding module of the decoding module, the cell state and the hidden state at any time t are represented by $C^{(t)}$ and $h^{(t)}$, respectively. In the encoding stage, the DGLSTM model performs state update according to the following formula:

$$\begin{cases} g_e^{(t)} = \sigma(W_g^e h^{(t-1)} + U_g^e(f_\theta * X^{(t)}) + b_g^e), \\ i_e^{(t)} = \sigma(W_i^e h^{(t-1)} + U_i^e(f_\theta * X^{(t)}) + b_i^e), \\ o_e^{(t)} = \sigma(W_o^e h^{(t-1)} + U_o^e(f_\theta * X^{(t)}) + b_o^e), \\ a_e^{(t)} = \tanh(W_a^e h^{(t-1)} + U_a^e(f_\theta * X^{(t)}) + b_a^e), \\ C^{(t)} = g_e^{(t)} \otimes C^{(t-1)} + i_e^{(t)} \otimes a_e^{(t)}, \\ h^{(t)} = o_e^{(t)} \otimes \tanh(C^{(t)}), \end{cases} \quad (10)$$

where σ and \tanh denote the sigmoid activation function and hyperbolic tangent activation function, respectively; $g_e^{(t)}$ is the proportion of cell information from the previous moment to the current moment; $i_e^{(t)}$ is the ratio of the previous hidden information and the current input information stored in the cell state; $o_e^{(t)}$ indicates the retention ratio of the updated cell state information through the tanh activation function; $W_g^e, U_g^e, b_g^e, W_i^e, U_i^e, b_i^e, W_o^e, b_o^e, U_o^e, W_a^e, U_a^e, b_a^e$ are neural network parameters at the encoding stage; and f_θ is the convolution filter with parameter θ .

In the model training stage, each sample's actual output is known, but in the online prediction stage, it is necessary to input a sequence to output the entire predicted sequence. Each time a forecasting value is generated, time advances one step further. When the last predicted value is obtained, the whole sequence generation ends. To predict the value at time t , the model needs to use the output value at time $t-1$ as the input. However, in the online prediction process, we cannot obtain the real output of the previous step and can only use the estimated output at an earlier time as the next time. If the predicted value error at the previous moment is large, we use the predicted value to estimate the value at the next moment, which will cause the prediction error to become larger and larger.

A sampling mechanism is proposed in [26] to reduce the gap between model training and online prediction. In the model training phase, the real value and predicted value at the previous moment are randomly selected as the input for the next moment. The real value is no longer wholly used as the input at the next moment through the random sampling mechanism. The actual value and model's output are, respectively, selected with probabilities p and $1-p$. It should

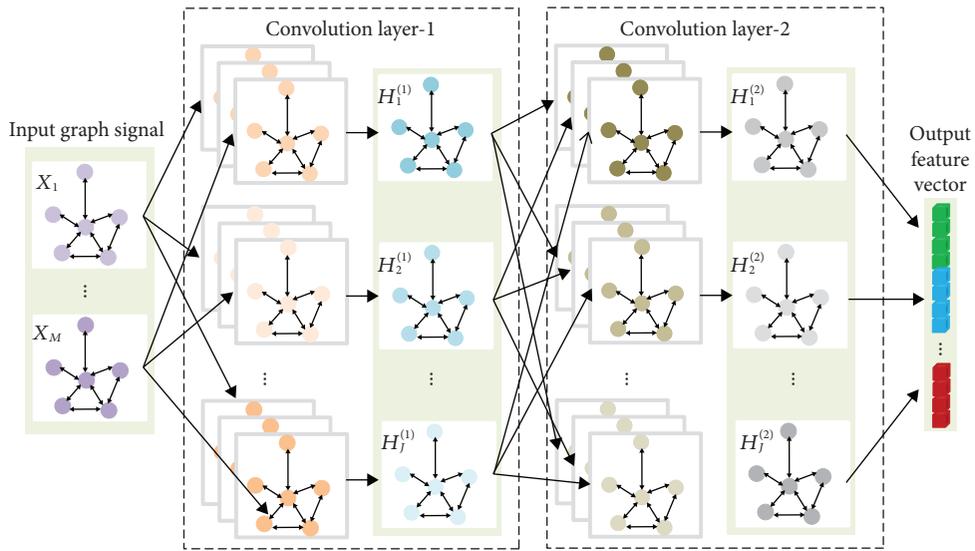


FIGURE 3: Schematic diagram of the feature representation of the graph network using two-layer convolution.

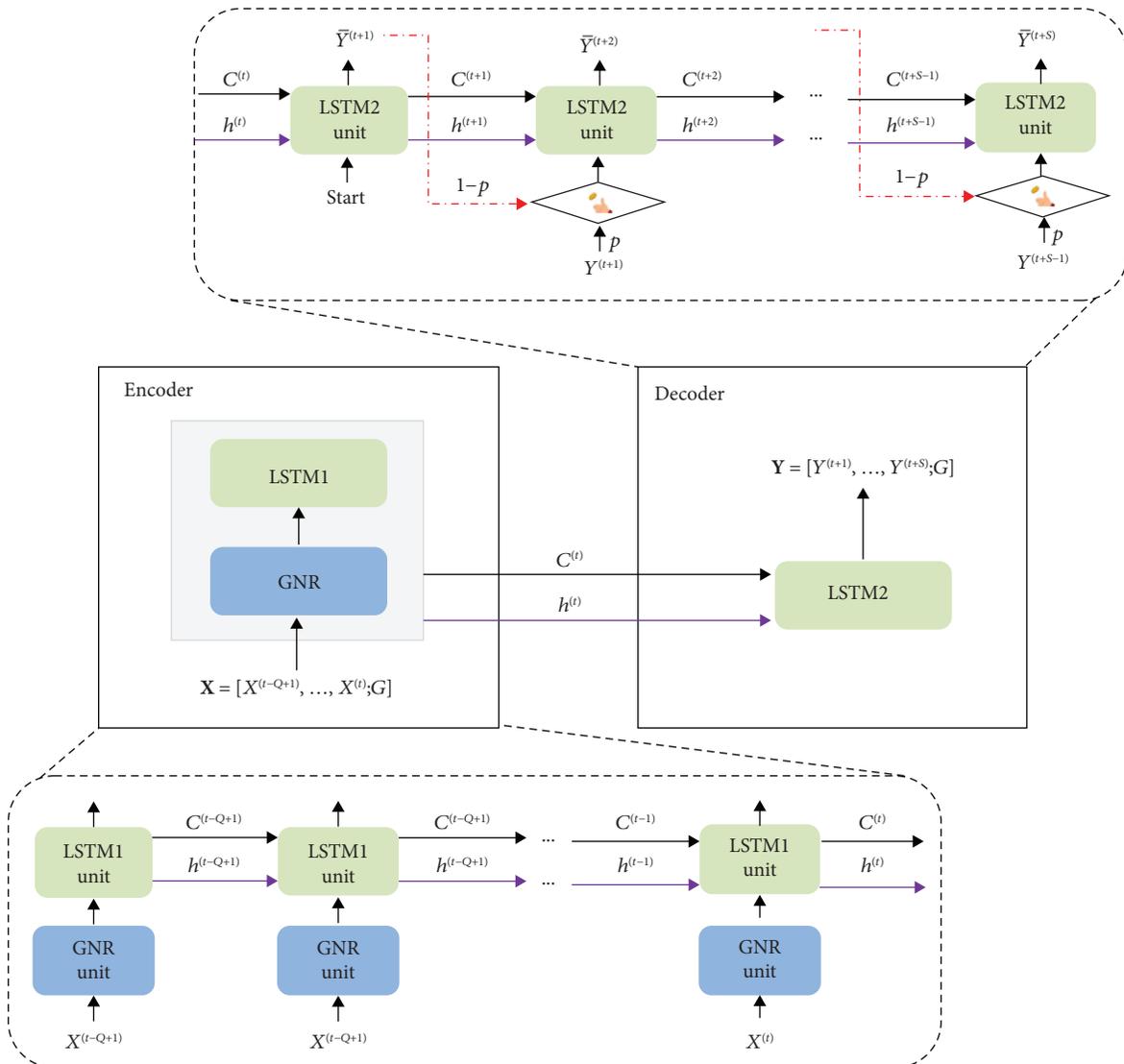


FIGURE 4: Network structure of the SS-LSTM prediction model.

be noted that p changes during the training process, just like the learning rate. Since the initial network training is insufficient, try to use the real value so that p should be as large as possible. As the training progresses, the model training becomes more and more sufficient. At this time, the value of p should also be reduced accordingly. The model's output is chosen as much as possible so that the model training and online prediction can be consistent. Therefore, under the concept of the sampling mechanism, the decoding stage updates the state according to the following formula:

$$\begin{cases} g_d^{(t)} = \sigma\left(W_g^d h^{(t-1)} + U_g^d\left(\text{rand}\left\{Y^{(t)}, \bar{Y}^{(t)}\right\}\right) + b_g^d\right), \\ i_d^{(t)} = \sigma\left(W_i^d h^{(t-1)} + U_i^d\left(\text{rand}\left\{Y^{(t)}, \bar{Y}^{(t)}\right\}\right) + b_i^d\right), \\ o_d^{(t)} = \sigma\left(W_o^d h^{(t-1)} + U_o^d\left(\text{rand}\left\{Y^{(t)}, \bar{Y}^{(t)}\right\}\right) + b_o^d\right), \\ a_d^{(t)} = \tanh\left(W_a^d h^{(t-1)} + U_a^d\left(\text{rand}\left\{Y^{(t)}, \bar{Y}^{(t)}\right\}\right) + b_a^d\right), \\ C^{(t)} = g^{(t)} \otimes C^{(t-1)} + i^{(t)} \otimes a^{(t)}, \\ h^{(t)} = o^{(t)} \otimes \tanh(C^{(t)}) \end{cases} \quad (11)$$

where $\text{rand}\left\{Y^{(t)}, \bar{Y}^{(t)}\right\}$ denotes a selection function that chooses a value from the actual value and the estimated value with a certain probability. The meaning of other variables is similar to the encoding stage.

Given C training samples $\{(\mathbf{X}_i, \mathbf{Y}_i): i = 1, 2, \dots, C\}$, where $\mathbf{X}_i = (X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(Q)})^T \in R^{P \times M \times Q}$ and $\mathbf{Y}_i = (Y_i^{(1)}, Y_i^{(2)}, \dots, Y_i^{(S)})^T \in R^{P \times N \times S}$ represent the i -th input and output sequences, respectively. For discrete-state problems, such as classification, the maximum probability of the sample category [26] or the cross-entropy [45] is usually utilized as the training loss function. For continuous value prediction problems, however, the square error is generally used as the loss function. Airport delay prediction belongs to a type of continuous problem, and the loss function is as follows:

$$\text{loss} = \sum_{i=1}^C \sum_{t=1}^S \left(\left\| \bar{Y}_i^{(t)} - Y_i^{(t)} \right\| \right), \quad (12)$$

where $\bar{Y}_i^{(t)}$ is the output of the i -th sample at time t . Taking the minimization of (12) as the optimization goal, we adopt the small-batch stochastic gradient descent method to solve parameters in the model.

5. Case Analysis

This section takes the US airport network as a case to verify the prediction results of the DGLSTM delay prediction model from different perspectives. First, we describe the experimental data and determine the best parameters of the model through experiments. Then, from the aspect of prediction duration, the performance and robustness of the DGLSTM model are analyzed. Furthermore, the DGLSTM method is compared with mainstream delay prediction

methods to verify its superiority. Finally, the spatial dependence and temporal dependence of the DGLSTM model are discussed. The mean absolute error (MAE) and mean relative error (MRE) are used to evaluate the model's prediction accuracy. The calculation formulas of MAE and MRE are as follows:

$$\begin{aligned} \text{MAE} &= \frac{1}{n} \sum_{i=1}^n |e_i - \bar{e}_i|, \\ \text{MRE} &= \frac{1}{n} \sum_{i=1}^n \frac{|e_i - \bar{e}_i|}{e_i}, \end{aligned} \quad (13)$$

where e_i represents the actual delay time, \bar{e}_i represents the predicted delay value, and n represents the total number of predicted values.

The experiment in this section is implemented by Python programming; the running environment is Intel(R) Core(TM) i5-8300H CPU @2.30 GHz processor, 96 GB running memory, 64-bit Windows 10 operating system.

5.1. Data Description. The experimental data in this paper come from the Bureau of Transportation Statistics (BTS), which provides flight operation data at major US airports (<https://www.bts.gov/>). Data from January 1, 2015, to December 31, 2018, are used, including 325 airports and 2,412,598 pieces. In the experiments of this paper, we predict the average flight delay of each airport in each hour of the next 10 hours. Therefore, one day is divided into 24 time intervals in hours. We count the average delay time per hour for each airport in hours and obtain 35,040 pieces of data. Each piece of data contains six attributes: time, airport name, departure delay, arrival delay, departure flight volume, and arrival flight volume. The input signal features are departure delay, arrival delay, departure flight volume, and arrival flight volume. The output signal feature is the departure delay. The dataset is divided into a training set, a validation set, and a test set according to the ratio of 8 : 1 : 1. To construct the airport graph network, we obtain the coordinate position of each airport and the flight distance airports, calculate the weighted adjacency matrix using equations (4)–(6), and perform normalization processing.

5.2. Experimental Setup. The DGLSTM model proposed in this paper needs to determine the graph network representation parameters and the LSTM network structure parameters. The graph network parameters include the weighting parameter α and diffusion step K . The weight parameter is selected from 0 to 1, with a step length of 0.05. The diffusion step length is selected from 1 to 6, with a step length of 1. In this paper, the optimal parameters are determined by the exhaustive method, and the optimal weight parameters and diffusion step length are 0.45 and 3, respectively. Regarding the LSTM network structure, it is necessary to determine the input layer's size, the number of hidden layers, and the number of corresponding neurons. The number of hidden layers has a more significant impact

on prediction accuracy. Too few layers cannot fully represent the features in the training dataset; on the contrary, too many neurons will increase the complexity of the model and easily overfit. For the input layer, the time interval determines the input layer structure, with k hours of history as the input. For the number of hidden layers, the selection range of the number of layers is greater than or equal to 1 and less than or equal to 7. The number of neurons in each layer is selected from the set [40, 80, 160, 256, 320, 512, 640, 768, 1000]. In the learning phase, the number of iterations is also significant. If the number of iterations is too large, the model will overfit. The maximum number of iterations is set to 10,000. This article uses a random method to select the value from each parameter's set, conduct 1000 experiments, and select the best configuration parameters. Under the data scale and server configuration used in the paper, the offline training model takes 4 hours and 12 minutes, and the online prediction takes 3 minutes and 25 seconds.

Next, we test the sensitivity of the input sequence length and diffusion step length. Figure 5 shows the experimental results of these two parameters. When the input sequence's length is 5, the prediction error reaches the lowest value, indicating that increasing the input sequence cannot improve the model's performance. Figure 5(b) shows the variation of MRE and MAE with diffusion step length. As the diffusion step size increases, the error first decreases rapidly and then slightly increases. When the diffusion step is 3, the model performance is optimal. Experimental results show that a larger diffusion step size enables the model to capture a broader range of spatial dependencies at the expense of increased learning complexity.

5.3. Experimental Results and Analysis. This section will evaluate the model's predictive performance from different perspectives and compare it with mainstream methods to verify its superiority.

The advantage of DGLSTM lies in its long-term memory ability. It uses an internal loop method to achieve multistep simultaneous prediction. In model training, a random sampling mechanism is introduced to improve the accuracy of online prediction. To examine the prediction effect of different prediction durations, Figure 6 shows the box plot of all test samples' MAE values. It can be seen from the figure that when predicting the delay of one hour in the future, the corresponding MAE is about 5, and the predicted fluctuation range is relatively small. As the length of prediction time increases, the prediction error gradually increases, and the magnitude of the prediction error gradually increases. It was further found that when the predicted duration exceeds 7 hours, the median value of MAE exceeds 10. The experimental results show that, with the increase of prediction time, the model's effect gradually deteriorates, and the fluctuation range of the prediction results is larger.

Figure 7 shows the prediction performance of DGLSTM at various airports. Color of the ball indicates the airport's delay level, and the size of the ball means the MAE value and MRE value. It can be seen from the figure that, for airports

with high delays, the MAE value is more significant, while the MRE value is smaller. The reason is that the higher the delay, the more busy the airport, and the reasons for the delay of this type of airport flight are complicated. Once a flight delay of this type of airport occurs, the spread of the delay will be severe, which will affect subsequent flights and related airports. For airports with a larger spherical radius and lighter colors, through the analysis of these airports' flight plans, it is found that most of them are airports with a small number of flights. For airports with fewer flights of this type, flight delays have relatively large randomness. It is generally challenging to capture periodic statistical laws from the data, so the prediction error is relatively low. Compared with high-delay and low-delay airports, the overall prediction accuracy of medium-delay airports is higher.

There are often significant differences between airports, and the geographic location of an airport largely determines how busy the airport is. An important indicator to measure the busyness of an airport is the number of flights. The larger the number of flights, the busier the airport. Figure 8 shows the relationship between MRE and MAE on flight volume. Figure 8(a) shows the relationship between flight volume and MAE. It shows that most airports in the United States have less than 10,000 flights per month. This type of airport has low delays and forecast errors between 2 and 6 minutes, but MRE fluctuates greatly. Figure 8(b) shows the relationship between flight volume and MRE. Airports with a large number of flights have an MRE between 0.05 and 0.1, while for airports with a small number of flights, the MRE value can reach up to 0.3. The possible reason is that, in airports with small flight volume, the regular operation of flights is often affected by neighboring airports or delayed spreading factors, which has greater randomness. For airports with a large number of flights, the MRE value and volatility are small. The result is that, in airports with a large number of flights, flight delays are normal. From historical operating data, the underlying delay laws can be dug out.

Figure 9 shows the delay prediction results for typical days at Atlanta Airport (ATL) and Atlantic City Airport (ACY). ATL belongs to an airport with relatively severe delays, while ACY belongs to an airport with relatively small delays. Figure 9(a) demonstrates the confidence interval of the historical delay of ATL. It can be seen from the figure that the predicted value's overall trend is in line with the real value. Figure 9(b) gives the experimental result of Atlantic City Airport. Between 1 am and 6 am, the delay is zero because there are no planes scheduled for this period. Compared with ATL, the overall delay level of ACY is lower, but the maximum delay point's prediction error is more significant. The main reason is that the number of flights is small, and the delay is more random.

To verify the superiority of this method, DGLSTM is compared with LSTM [25], random forest (RF) [16], BP neural network (BPNN) [20], and Markov method. LSTM only used information related to delays at the current airport. BPNN considered the impact of the airports connected by flights but ignored the impact of neighboring airports. RF considered the characteristics of the delay state of the entire

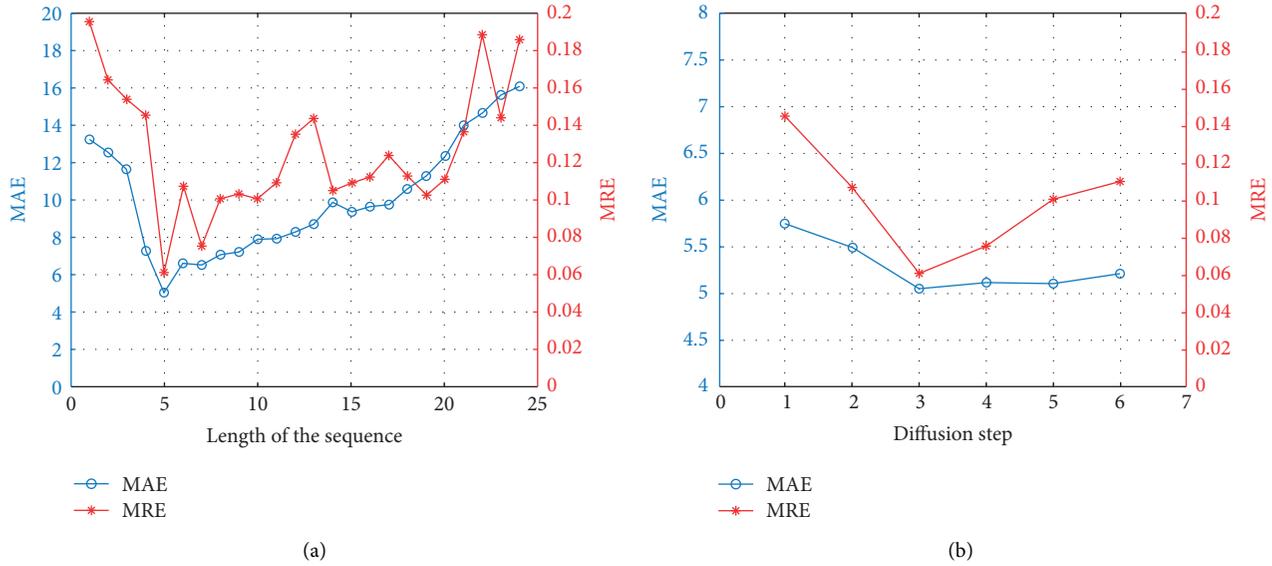


FIGURE 5: Sensitivity analysis of input sequence length and diffusion step length.

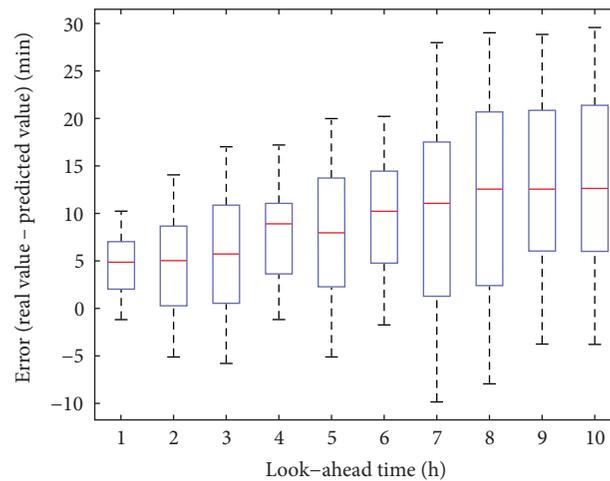


FIGURE 6: Prediction error of DGLSTM as a function of varying look-ahead times.

airport network, thereby taking into account the spatial information to a certain extent, but does not measure the associated airport's impact in a refined manner. The Markov method is a widely used method in many prediction problems. Figure 10 shows the prediction errors of different methods. The Markov method's prediction performance is poor, MRE is about 0.4, and MAE is more than 20 minutes. When the random forest method is used for forecasting, as the forecasting time becomes longer, the forecasting performance gradually decreases. The overall effect of the BP neural network is better than the previous two types of models. Although the prediction accuracy fluctuates, the MRE is within 0.2. The DGLSTM method inherits the long-term memory of the LSTM method, and the overall effect is better than the previous three methods. In general, DGLSTM has the highest prediction accuracy. MRE is 0.01–0.03 lower than LSTM, 0.05–0.07 lower than the BP

neural network, 0.0–0.12 lower than the random forest, and 0.3 lower than Markov average. Although the accuracy gradually decreases with the increase of prediction time, it is still better than the mainstream methods. The excellent performance of DGLSTM benefits from inheriting the long-term memory characteristics of LSTM and the global characteristics of graph neural networks.

5.4. Model Discussion. To explore the influence of the weighted adjacency matrix on the performance of DGLSTM, the spatial weighted adjacency matrix, the demand weighted adjacency matrix, and the combined weighted adjacency matrix were discussed separately. (a) The DGLSTM model that only uses the spatial weighted adjacency matrix can better characterize the delay propagation characteristics between adjacent airports in space. This model is referred to

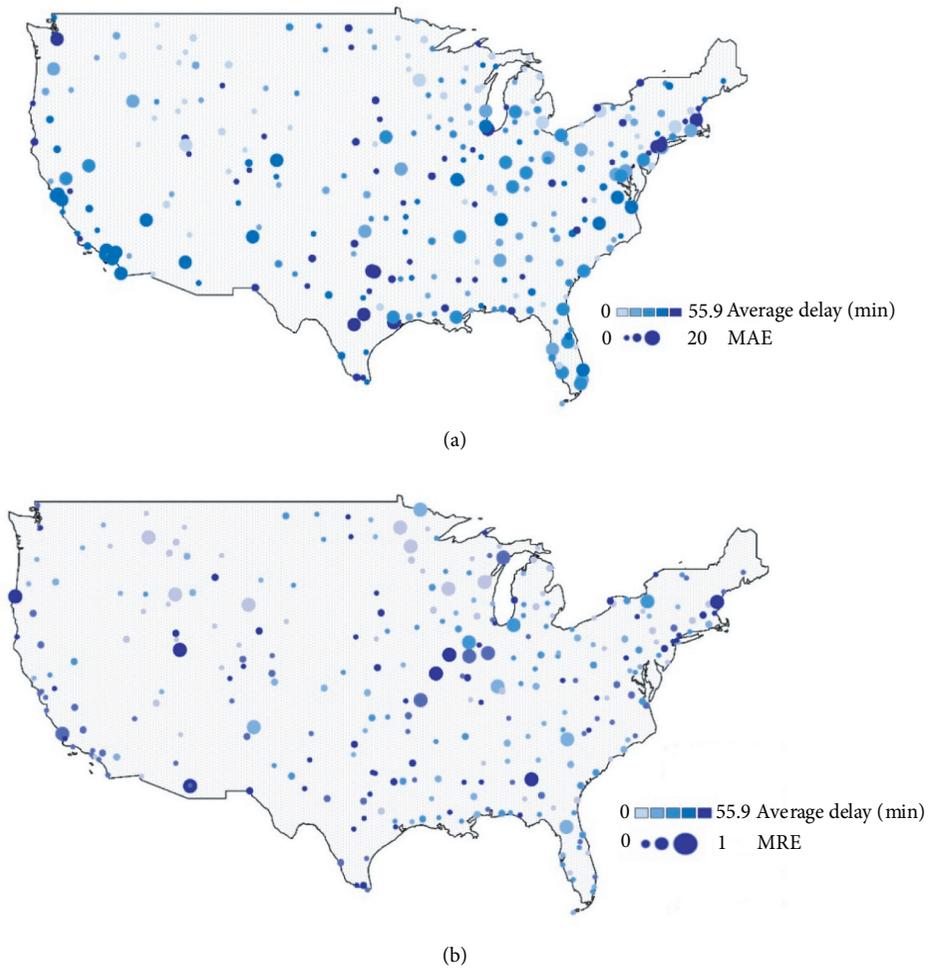


FIGURE 7: The prediction errors of airports with different delay levels. Color of the ball indicates the degree of airport delay, and the size of the ball means the airport's MAE and MRE values. (a) Relationship between delay and MAE. (b) Relationship between delay and MRE.

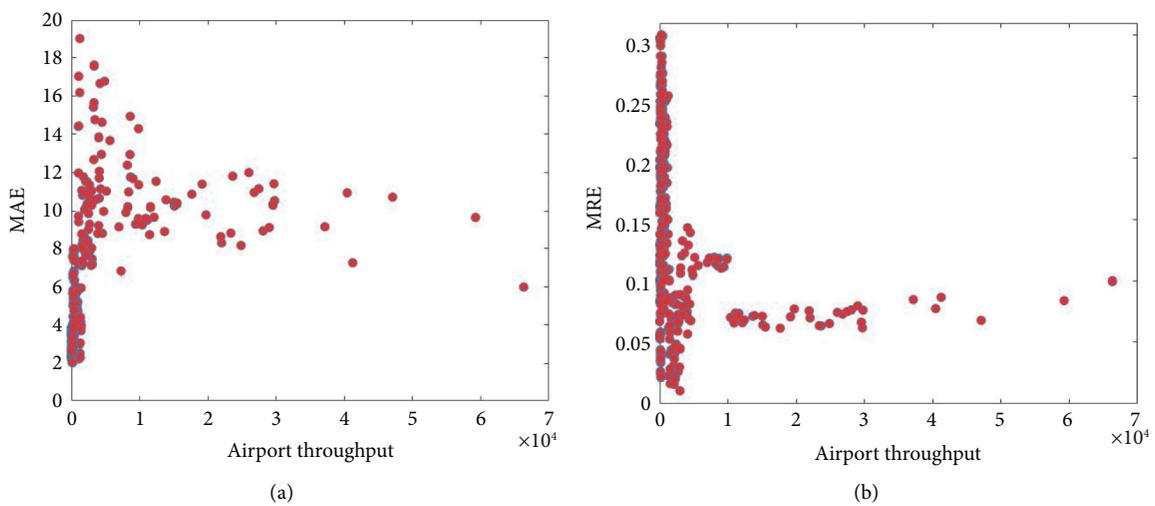


FIGURE 8: Relationship between the forecast error and the flight volume.

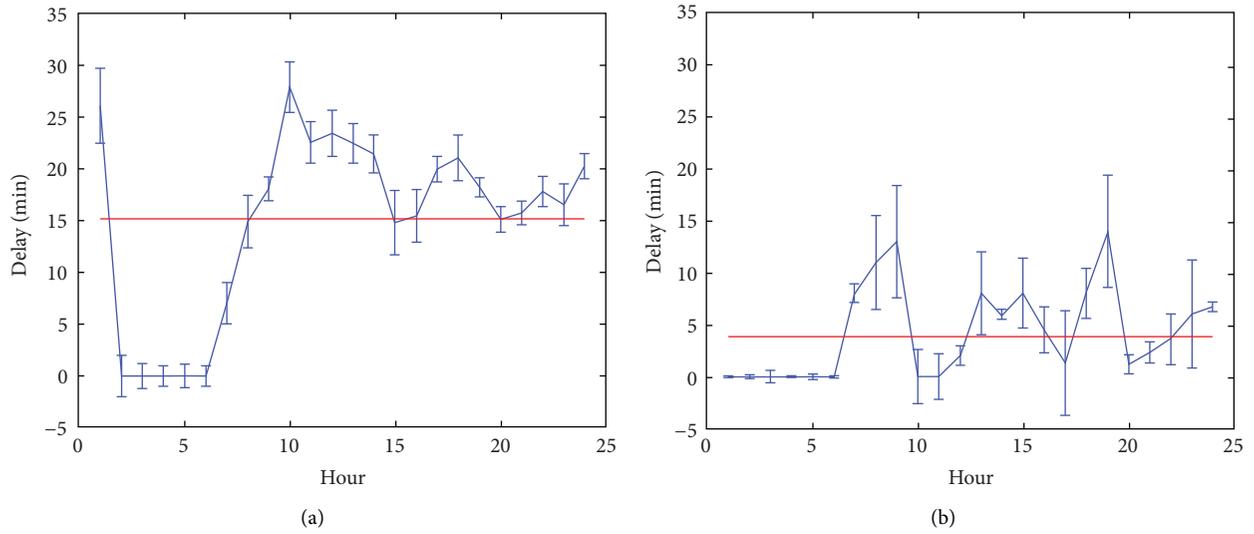


FIGURE 9: Error bars of (a) Atlanta Airport (ATL) and (b) Atlantic City Airport (ACY).

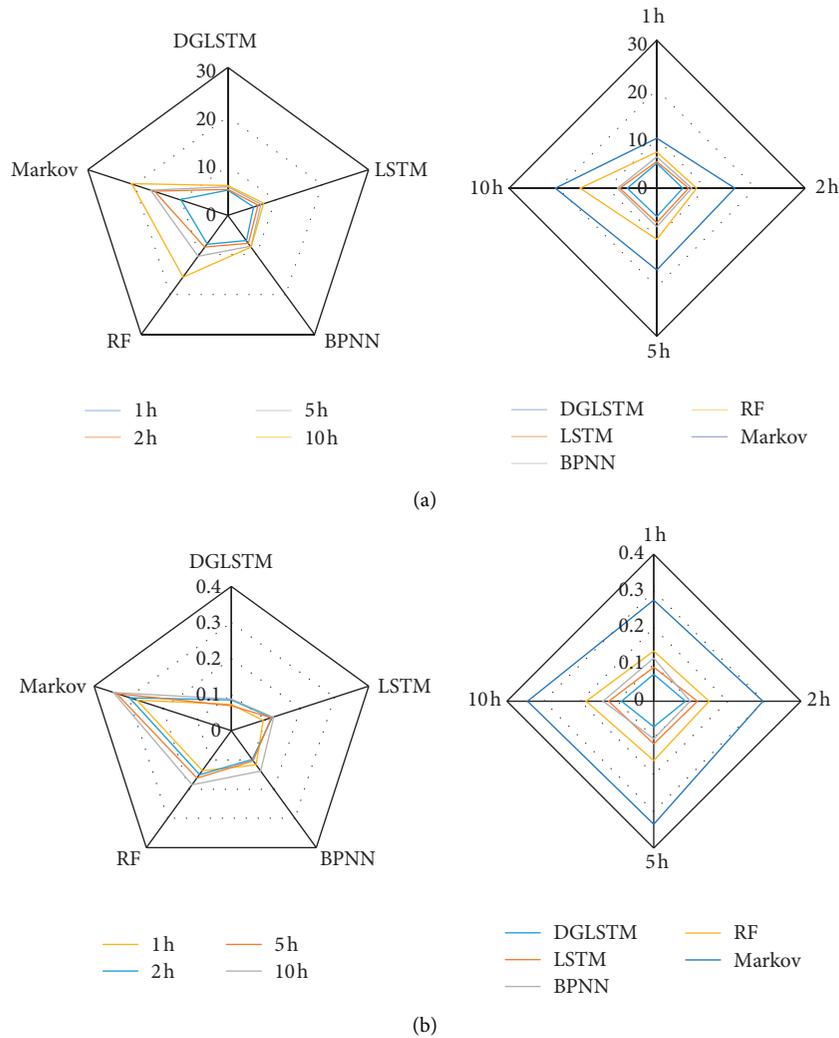


FIGURE 10: Comparison of experimental results of different methods. (a) MAE. (b) MRE.

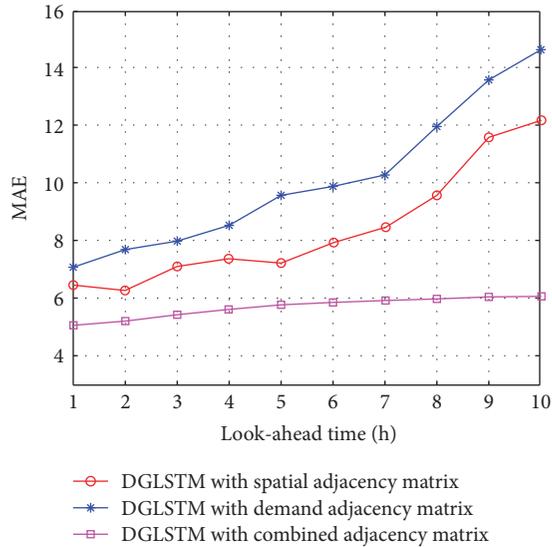


FIGURE 11: Influence of the adjacency matrix on the prediction model.

as “DGLSTM with spatial weighted adjacency matrix”; (b) the DGLSTM model that only uses the demand-weighted adjacency matrix can well capture the delay characteristics propagated through the route. This model is referred to as “DGLSTM with demand weighted adjacency matrix”; (c) the DGLSTM model using a combined weighted adjacency matrix, taking into account the delays propagated by the space distance and the route, is denoted as “DGLSTM with combined weighted adjacency matrix.”

Figure 11 shows the experimental results of DGLSTM using different weighted adjacency matrices. It can be seen from the figure that the effect of DGLSTM with the spatial weighted adjacency matrix is better than that of DGLSTM with the demand weighted adjacency matrix. It suggests that the impact between adjacent airports is more significant than that of airports connected by flights. The reason behind this may be that neighboring airports share airspace resources, and the airspace resources are limited in capacity. A delayed flight at an airport will postpone using airspace resources, resulting in delays due to flow control at adjacent airports. It is further found that the DGLSTM model that only considers distance or demand has a rapid increase in MAE as the forecast duration increases, especially when the forecast duration exceeds 5 hours, the error level increases rapidly. However, as the forecast time becomes longer, the MAE of DGLSTM with combined weighted adjacency matrix grows slowly, indicating that comprehensive consideration of distance and demand factors can better capture flight delay characteristics and improve the model’s forecasting performance.

DGLSTM uses sequence-to-sequence LSTM to capture temporal features and uses a sampling mechanism in the decoding stage to improve the model’s robustness. This experiment will examine the effect of the sampling mechanism on the model performance. Figure 12 shows the experimental results of DGLSTM with and without a sampling mechanism. The prediction effect of the model

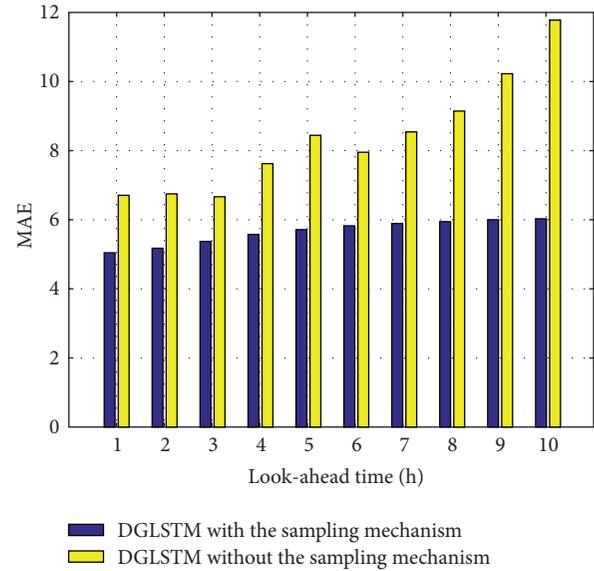


FIGURE 12: Influence of the sampling mechanism on the prediction model.

using the sampling mechanism is better than that of not using the sampling mechanism. As the prediction time increases, its superiority becomes more apparent. When the length forecast time is 1 hour, the gap between them is within 2 minutes. As the length of forecast time increases, the difference between the MAE values of the two is about 6 minutes when the forecast time is 10 hours. The main reason is that when the model using the sampling mechanism is trained, it is randomly selected from the estimated value at the previous moment and the real value. As the number of iterations increases, the probability of choosing from the estimated value gradually increases without using the sampling mechanism. However, instead of using a mechanism to train the model, real input is used, resulting in poor robustness in online prediction.

6. Conclusions

This paper proposes a deep DGLSTM delay prediction model from the perspective of airport networks. Regarding the airport network as a directed graph network with an airport as a node, a spatial distance weighted adjacency matrix and a demand weighted adjacency matrix are constructed, and the two are integrated to obtain a combined weighted adjacency matrix. On this basis, we use diffusion convolution to characterize the airport graph network to capture the delay propagation characteristics between airports. Furthermore, it is proposed to use a sequence-to-sequence LSTM model to mine time-domain features, including encoding and decoding modules. The encoding module takes the output of the graph network representation module as the input, and the decoding module takes the output of the encoding module as the input. In the model training process, we use the sampling mechanism to select between the estimated value and the real value according to a certain probability to improve the robustness of the online

prediction stage. The experiment takes 325 major airports in the United States as an example, uses historical operating data of all flights from 2015 to 2018, and uses MAE and MRE to evaluate the model's performance. Experimental results show that the method proposed in this paper has higher prediction accuracy and robustness and is better than the current mainstream methods. In future research, we will try to use some advanced feature representation or prediction methods to improve model prediction accuracy [46, 47] and parallel computing methods to speed up model training [48].

Data Availability

The data used to support the findings of this study come from the Bureau of Transportation Statistics, USA (<https://www.bts.gov/>). Researchers can download these data from this platform.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (nos. 62076126 and 62006041) and the Fundamental Research Funds for the Central Universities (no. NS2018044).

References

- [1] VariFlight, *Global Airport and Airline Punctuality Report*, VariFlight, Hefei, China, 2020.
- [2] K. G. Zografos, M. A. Madas, and K. N. Androutsopoulos, "Increasing airport capacity utilisation through optimum slot scheduling: review of current developments and identification of future needs," *Journal of Scheduling*, vol. 20, no. 1, pp. 3–24, 2017.
- [3] P. Pellegrini, T. Bolić, L. Castelli, and R. Pesenti, "SOSTA: an effective model for the simultaneous optimisation of airport slot allocation," *Transportation Research Part E: Logistics and Transportation Review*, vol. 99, pp. 34–53, 2017.
- [4] C. S. Bosson and D. Sun, "Optimization of airport surface operations under uncertainty," *Journal of Air Transportation*, vol. 24, no. 3, pp. 84–92, 2016.
- [5] R. Arnaldo Scarpel and L. C. Pelicioni, "A data analytics approach for anticipating congested days at the São Paulo International Airport," *Journal of Air Transport Management*, vol. 72, pp. 1–10, 2018.
- [6] A. S. Estes and M. O. Ball, "Equity and strength in stochastic integer programming models for the dynamic single airport ground-holding problem," *Transportation Science*, vol. 54, no. 4, pp. 944–955, 2020.
- [7] N. Kenan, A. Jebali, and A. Diabat, "The integrated aircraft routing problem with optional flights and delay considerations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 118, pp. 355–375, 2018.
- [8] Y. Tu, M. O. Ball, and W. S. Jank, "Estimating flight departure delay distributions-A statistical approach with long-term trend and short-term pattern," *Journal of the American Statistical Association*, vol. 103, no. 481, pp. 112–125, 2008.
- [9] E. R. Mueller and G. B. J. A. J. Chatterji, "Analysis of aircraft arrival and departure delay characteristics," in *Proceedings of the AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, p. 5866, Los Angeles, CA, USA, October 2002.
- [10] B. Sridhar and N. Y. Chen, "Short-term national airspace system delay prediction using weather impacted traffic index," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 657–662, 2009.
- [11] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara, "A review on flight delay prediction," 2017, <https://arxiv.org/abs/1703.06118>.
- [12] A. Klein, C. Craun, and R. S. Lee, "Airport delay prediction using weather-impacted traffic index (WITI) model," in *Proceedings of the 29th Digital Avionics Systems Conference*, Salt Lake City, UT, USA, October 2010.
- [13] Z. J. Hanley, *Delay Characterization and Prediction in Major US Airline Networks*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2015.
- [14] Á. Rodríguez-Sanz, F. G. Comendador, R. A. Valdés, J. Pérez-Castán, R. B. Montes, and S. C. Serrano, "Assessment of airport arrival congestion and delay: prediction and reliability," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 255–283, 2019.
- [15] J. Ding and H. J. P. P. Li, "The forecasting model of flight delay based on DMT-GMT model," *Physics Procedia*, vol. 33, pp. 395–402, 2012.
- [16] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 231–241, 2014.
- [17] T. Zhou, Q. Gao, X. Chen, and Z. Xun, "Flight delay prediction based on characteristics of aviation network," in *Proceedings of the MATEC Web of Conferences. EDP Sciences*, Chicago, IL, USA, November 2006.
- [18] X. Yang, Y. Zou, J. Tang, J. Liang, and M. Ijaz, "Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models," *Journal of Advanced Transportation*, vol. 2020, Article ID 9628957, 16 pages, 2020.
- [19] W. Zeng, Z. Quan, Z. Zhao, C. Xie, and X. Lu, "A deep learning approach for aircraft trajectory prediction in terminal airspace," *IEEE Access*, vol. 8, pp. 151250–151266, 2020.
- [20] S. Khanmohammadi, S. Tutun, and Y. J. P. C. S. Kucuk, "A New multilevel input layer artificial neural network for predicting flight delays at JFK airport," *Procedia Computer Science*, vol. 95, pp. 237–244, 2016.
- [21] R. Henriques and I. Feiteira, "Predictive modelling: flight delays and associated factors, hartsfield-Jackson Atlanta International airport," *Procedia Computer Science*, vol. 138, pp. 638–645, 2018.
- [22] K. R. Chandramouleeswaran, D. Krzemien, K. Burns, and H. T. Tran, "Machine learning prediction of airport delays in the US air transportation network," in *Proceedings of the 2018 Aviation Technology, Integration, and Operations Conference, AIAA AVIATION Forum*, American Institute of Aeronautics and Astronautics, Los Angeles, CA, USA, 2018.
- [23] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: a deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203–221, 2019.
- [24] Z. Li, H. Chen, J. Ge, and K. Ning, *An Airport Scene Delay Prediction Method Based on LSTM*, Springer, Berlin, Germany, 2018.

- [25] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1–6, Sacramento, CA, USA, 2016.
- [26] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 1171–1179, Montreal, Canada, December 2015.
- [27] G. M. Scarselli and A. C. Tsoi, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [28] T. D. Li and M. Brockschmidt, "Gated graph sequence neural networks," 2015, <https://arxiv.org/abs/1511.05493>.
- [29] K. Z. Dai and B. Dai, "Learning steady-states of iterative algorithms over graphs," in *Proceedings of the International Conference on Machine Learning*, pp. 1114–1122, Stockholm, Sweden, July 2018.
- [30] Y. Z. Hamilton and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1024–1034, Long Beach, CA, USA, December 2017.
- [31] B. X. Defferrard, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, pp. 3844–3852, MIT Press, Cambridge, MA, USA, 2016.
- [32] W. M. Kipf, "Semi-supervised classification with graph convolutional networks," 2017, <https://arxiv.org/abs/1609.02907>.
- [33] K. S. I. Parisot and E. Ferrante, "Disease prediction using graph convolutional networks: application to Autism Spectrum Disorder and Alzheimer's disease," *Medical Image Analysis*, vol. 48, pp. 117–130, 2018.
- [34] Y. H. Yu and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," 2018, <https://arxiv.org/abs/1709.04875>.
- [35] G. C. Hu and B. Yang, "Stochastic weight completion for road networks using graph convolutional networks," in *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE)*, Beijing, China, April 2019.
- [36] N. S. K. Shuman and P. Frossard, "The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains," 2013, <https://arxiv.org/abs/1211.0053>.
- [37] F. K. Li and Z. Wang, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference*, pp. 1695–1704, London, UK, August 2018.
- [38] Z. W. Bruna and A. Szlam, "Spectral networks and locally connected networks on graphs," 2014, <https://arxiv.org/abs/1312.6203>.
- [39] Y. R. Li and C. Shahabi, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," 2018, <https://arxiv.org/abs/1707.01926>.
- [40] C. Xu, J. Ji, and P. Liu, "The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 47–60, 2018.
- [41] F. Althé and A. De La Fortelle, "An LSTM network for highway trajectory prediction," in *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems*, pp. 353–359, Yokohama, Japan, 2018.
- [42] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
- [43] X. Li, L. Peng, X. Yao et al., "Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation," *Environmental Pollution*, vol. 231, pp. 997–1004, 2017.
- [44] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–91, 2016.
- [45] B. V. L. Kline, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing & Applications*, vol. 14, no. 4, pp. 310–318, 2005.
- [46] X. Chen, S. Wang, C. Shi, H. Wu, J. Zhao, and J. Fu, "Robust ship tracking via multi-view learning and sparse representation," *Journal of Navigation*, vol. 2019, no. 1, pp. 176–192, 2018.
- [47] X. Chen, X. Xu, Y. Yang, H. Wu, J. Tang, and J. Zhao, "Augmented ship tracking under occlusion conditions from maritime surveillance videos," *IEEE Access*, vol. 8, pp. 42884–42897, 2020.
- [48] J. Chen, K. Li, H. Rong, K. Bilal, K. Li, and P. S. Yu, "A periodicity-based parallel time series prediction algorithm in cloud computing environments," *Information Sciences*, vol. 496, pp. 506–537, 2018.